We claim:

1. A computerized method for providing extended functionality to an extensible object in an object model comprising:

locating a package for the extended functionality;

obtaining an extension object for the extended functionality from the package; and

directing references to the extended functionality to the extension object.

2. The method of claim 1, wherein locating the package comprises:

searching for an entry for the extension in an extension database.

3. The method of claim 1, wherein obtaining the extension object comprises:

creating the extension object when the extended functionality is first referenced; and

locating the extension object when the extended functionality is subsequently referenced.

4. The method of claim 3, further comprising storing the extension object in a cache memory so that the extension object is located in cache memory upon subsequent references to the extended functionality.

5. The method of claim 3, wherein creating the extension object further comprising:

obtaining an extension provider object; and

specifying parameters to the extension provider object to create the extension object.

~~6. The method of claim 5, wherein obtaining an extension object comprises:~~

~~creating the extension provider object if it does not exist; and~~

~~locating the extension provider object if it does exist.~~

~~7. The method of claim 6, further comprising storing the extension provider object in a cache memory so that the extension provider is located in cache memory when it exists.~~

~~8. The method of claim 1, further comprising:~~

~~storing the package on a persistent storage; and~~

~~creating an entry in an extension database for the package to register the package.~~

~~9. The method of claim 5, further comprising:~~

~~obtaining the package from an external source.~~

~~10. The method of claim 1, wherein the elements are performed in the order recited.~~

SUB A1> ~~11. A computer-readable medium having stored thereon an entry for an extension in extension data structure comprising:~~

~~an extension identifier field containing data representing an identifier for the extension;~~

~~an extension name field containing data representing an external name for the extension identified by the extension identifier field ; and~~

a extension object identifier field containing data representing an extension package for the extension identified by the extension identifier field.

12. The computer-readable medium of claim 11, further comprising:

an object identifier field containing data representing an extensible object having functionality extended by the extension identified by the extension identifier field.

13. The computer-readable medium of claim 11, further comprising:

an internal name field containing data representing an internal name for the extension identified by the extension identifier field.

14. The computer-readable medium of claim 11, further comprising:

a friendly name field containing data representing a browsable name for the extension identified by the extension identifier field.

15. The computer-readable medium of claim 11, further comprising:

a description field containing data representing a textual description of the extension identified by the extension identifier field.

Sub B1

16. A computerized system comprising:

a processing unit;

a system memory coupled to the processing unit through a system bus;

a computer-readable medium coupled to the processing unit through a system bus;

~~an extensible object model executed from the computer-readable medium by the processing unit, wherein the extensible object model causes the processing unit to create an extension object from an extension package when a corresponding extension for an extensible object is first referenced and to delegate subsequence references to the extension to the extension object.~~

Sub C1 ~~17. The computerized system of claim 16, wherein the extensible object model further causes the processing unit to notify the extensible object when the extension object is deleted.~~

Sub E1 ~~18. The computerized system of claim 16, wherein the extensible object model further causes the processing unit to register the extension package in an extension database stored on the computer-readable medium.~~

~~19. The computerized system of claim 16, wherein the extensible object model further causes the processing unit to store the extension object in system memory when the corresponding extension is first referenced.~~

~~20. The computerized system of claim 16, wherein the extensible object model causes the processing unit to create the extension object from the extension package by causing the processing unit to create an extension provider object and causes the processing unit to create the extension object from the extension provider object.~~

21. The computerized system of claim 16, wherein the extensible object model further causes the processing unit to create an event filtering and sourcing object to handle events generated by the extension object.

~~22. A method of providing an extension to an extensible object in a run time environment comprising:~~

~~issuing, by the runtime environment, a GetAutomationObject call to an extension package;~~

~~proffering, by the extension package, an extension object that provides the extension as a result of the GetAutomationObject call;~~

~~associating, by the runtime environment, the extension object and the extensible object in an extension database; and~~

~~resolving, by the runtime environment, an invocation of the extension for the extensible object to the extension object through the extension database.~~

~~23. The method of claim 22, further comprising:~~

~~issuing, by the extension object, a NotifyDelete call to the extensible object when the extension object is being deleted.~~

~~24. A method of providing an extension to an extensible object in a run time environment comprising:~~

~~proffering, by an extension package, an extension provider object associated with the extension;~~

issuing, by the run time environment, a QueryInterface call to the extension provider object;

returning, by the extension provider object, an IExtensibleObjectProvider interface as a result of the QueryInterface call;

issuing, by the run time environment, a GetAutomationObject call to the IExtensibleObjectProvider interface;

proffering, by the extension provider object, an extension object, which services invocation of the extension, as a result of the GetAutomationObject call;

associating, by the runtime environment, the extensible object and the extension object in parent-child relationship; and

resolving, by the runtime environment, an invocation of the extension for the extensible object to the extension object.

25. The method of claim 24, further comprising:

calling, by the extension provider object, an IExtensibleObjectSite interface in the extensible object to obtain an identifier for the extensible object.

26. A method of providing an extension to an extensible object in a run time environment comprising:

storing, by a development environment, a GetObject method name in a fixed name space for the extensible object;

invoking, by the development environment, the extension for the extensible object using the GetObject method; and

~~binding, by the run time environment, an extension object for the extension to the extensible object by executing the GetObject method.~~

add 27

~~27. A computer-readable medium having stored thereon computer-executable components comprising:~~

~~an extensible object;~~

~~an extension database having an entry for an extension for the extensible object; and~~

~~an extension package having an interface for obtaining an extension object that provides the extension for the extensible object.~~

~~28. The computer-readable medium of claim 27, further comprising:~~

~~an extension provider object that proffers the extension object as a result of a call to the interface in the extension package.~~

add B2